100 Impediments

In Scrum, the ScrumMaster helps remove impediments. Certainly that's a challenge, but it can be a challenge to *recognize* impediments as well. With that in mind, here are some potential impediments to watch out for.

Organizational Structure

1. Gridlocked matrix management
2. Separate organizations for QA, Analysis, DBA, CM, ...
3. Lack of collocation

External Groups

4. Conflict between teams
5. Refusal to engage with a team
6. Distractions/ interruptions

Resources

7. Contractual risks
8. Fixed time, price, *and* scope
9. Unrealistic schedule
10. Unrealistic budget
11. Unrealistic estimates
12. Externally imposed estimates (for stories or projects)
13. Lack of configuration management
14. Inadequate hardware, systems, network

Policies

15. Compensation structure that works against teamwork
16. Mandated but ineffective processes
17. Mandated but ineffective tools

Process

18. Ineffective process
19. Unsure about process
20. Incompatible processes blended
21. Over-analysis up front
22. Insufficient analysis
23. Decisions locked in too early
24. No product backlog
25. ScrumMaster not doing their job
26. Team doesn't break down items well
27. Lack of tracking (burndown or other)
28. Team not able to demonstrate product

Sprints

29. Variable sprint length
30. Sprints extended (rather than timeboxed)
31. Sprint not closed to change

Management

32. Management style conflicts with Agile (e.g, command and control)
33. Management sends mixed message for support of Agile
34. Lack of sponsor or owner
35. Diverting people
36. Too many projects
37. Product not tied to business value
38. Imposed overhead (e.g., mandatory meetings)

Technical Risks

39. Performance
40. Gold-plating
41. Not potentially deployable
42. No system integration
43. Team not unified on design decisions
44. Too few tests
45. Not enough refactoring
46. Tests defined late
47. Infeasible requirements
48. Barely feasible requirements

100 Impediments

Team Agreements

49. No definition of done
50. Definition of done omits testing
51. Product Owner or other team member skips planning meeting
52. Team doesn't adjust when they're behind schedule
53. Retrospective yields no changes
54. Impediments not tracked
55. Impediments not addressed
56. No sprint backlog (or task list)
57. Team doesn't own sprint backlog

Quality

58. Unit tests inadequate
59. Poor refactoring
60. Tests defined late
61. Poor-quality code
62. Quality too low
63. Insufficient test automation

Product Risks

64. No Product Owner
65. PO unavailable
66. Multiple (disagreeing) Product Owners
67. No access to users
68. PO not splitting too-large stories
69. Wrong features
70. Poor user interface
71. Requirements thrashing
72. Unclear requirements
73. Market risks
74. Unclear vision
75. Missing stakeholders/ buy-in
76. Features added "just in case"
77. Backlog detailed too far in advance
78. Backlog not prioritized

People

79. Wrong skill-sets
80. Overspecialization
81. Wrong number of people
82. Bottlenecks
83. Slow on-boarding of new team members
84. Lack of ownership by team
85. Interpersonal conflict
86. Lack of trust

Skills

87. Teamwork
88. Analysis
89. Design
90. Implementation
91. Testing
92. Refactoring
93. Communication
94. Unsure how to use chosen technology

Environment/Tools

95. Ineffective tools for Product Owner, QA, or development
96. Slow or unreliable builds
97. Third-party components
98. Programming language
99. Lead time on hardware/software

100. ???

You probably noticed there aren't really 100 impediments in this list. That's because no closed list can capture everything that can be an impediment. However, with a list like this, or one of your own, you can sensitize yourself to the issues that slow down your team.